```
RelationOps: trait
  % Useful non-primitive operators on relations.
  assumes RelationBasics
  includes
    DerivedOrders (R, ⊆ for ≤, ⊇ for ≥,
                   ⊂ for <, ⊃ for >)
  introduces
    __ ∈ __, __ ∉ __: E, R → Bool
    set, dom, range, __⁺, __*: R → R
    __ ∩ __, __ ∘ __, __ - __, __ × __: R, R → R
    domRestrict, rangeRestrict, image: R, R → R
    skolem: E, R, R, E → E
  asserts
    ∀ e, e1, e2, e3: E, r, r1, r2: R
      e ∈ r == e ⟨ r ⟩ e;
      e ∉ r == ¬(e ∈ r);
      set(r) == r ∩ I;
      dom(r) == set(r ∘ ⊤);
      range(r) == set(⊤ ∘ r);
      e1 ⟨ r1 ∩ r2 ⟩ e2 == e1 ⟨ r1 ⟩ e2 ∧ e1 ⟨ r2 ⟩ e2;
      (e1 ⟨ r1 ⟩ e2 ∧ e2 ⟨ r2 ⟩ e3)
        ⇒ e1 ⟨ r1 ∘ r2 ⟩ e3;
      e1 ⟨ r1 ∘ r2 ⟩ e2
        ⇒ (e1 ⟨ r1 ⟩ skolem(e1, r1, r2, e2)
          ∧ skolem(e1, r1, r2, e2) ⟨ r2 ⟩ e2);
      r⁺ == r ∘ (r*);
      r* == I ∪ (r⁺);
      (r1 = I ∪ r2 ∧ r2 = r ∘ r1) ⇒
        ((r*) ⊆ r1 ∧ (r⁺) ⊆ r2);
      r1 - r2 == r1 ∩ (-r2);
      r1 × r2 == set(r1) ∘ ⊤ ∘ set(r2);
      r1 ⊆ r2 == r1 - r2 = ⊥;
      domRestrict(r1, r2) == r1 ∩ (r2 ∘ ⊤);
      image(r1, r2) == set(r1) ∘ r2;
      rangeRestrict(r1, r2) == r1 ∩ (⊤ ∘ r2)
  implies
    AbelianMonoid (⊤ for unit, ∩ for ∘, R for T),
    Distributive (∪ for +, ∩ for *, R for T),
    Distributive (∩ for +, ∪ for *, R for T),
    Idempotent (set, R),
    Monoid (I for unit, R for T),
    Lattice (R for T, ∪ for ⊔, ∩ for ⊓,
      ⊆ for ≤, ⊇ for ≥, ⊂ for <, ⊃ for >),
    PartialOrder (R, ⊆ for ≤, ⊇ for ≥, ⊂ for <,
      ⊃ for >)
```

```
∀ e: E, r, r1, r2: R
  e ∈ r == e ∈ set(r);
  -(r1 ∪ r2) == (-r1) ∩ (-r2);
  -(r1 ∩ r2) == (-r1) ∪ (-r2);
  (r1 ∘ r2)⁻¹ == (r2⁻¹) ∘ (r1⁻¹)
```

converts

```
  ∈, ∉, set, dom, range, __⁺, __*, __-__, ×,
  ∪, ∩, ∘, -:R→R, ⁻¹, ⊆, ⊇, ⊂, ⊃,
  domRestrict, image, rangeRestrict
```